F 13

```
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP        SSSSSSSSSSSS  YYY         YYY
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP        SSSSSSSSSSSS  YYY         YYY
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP        SSSSSSSSSSSS  YYY         YYY
UUU         UUU  EEE                    TTT        PPP         PPP  SSS              YYY         YYY
UUU         UUU  EEE                    TTT        PPP         PPP  SSS              YYY         YYY
UUU         UUU  EEE                    TTT        PPP         PPP  SSS              YYY         YYY
UUU         UUU  EEE                    TTT        PPP         PPP  SSS                 YYY   YYY
UUU         UUU  EEE                    TTT        PPP         PPP  SSS                 YYY   YYY
UUU         UUU  EEE                    TTT        PPP         PPP  SSS                 YYY   YYY
UUU         UUU  EEEEEEEEEEEE           TTT        PPPPPPPPPPPP        SSSSSSSSS           YYY
UUU         UUU  EEEEEEEEEEEE           TTT        PPPPPPPPPPPP        SSSSSSSSS           YYY
UUU         UUU  EEEEEEEEEEEE           TTT        PPPPPPPPPPPP        SSSSSSSSS           YYY
UUU         UUU  EEE                    TTT        PPP                          SSS        YYY
UUU         UUU  EEE                    TTT        PPP                          SSS        YYY
UUU         UUU  EEE                    TTT        PPP                          SSS        YYY
UUU         UUU  EEE                    TTT        PPP                          SSS        YYY
UUU         UUU  EEE                    TTT        PPP                          SSS        YYY
UUU         UUU  EEE                    TTT        PPP                          SSS        YYY
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT        PPP              SSSSSSSSSSSS           YYY
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT        PPP              SSSSSSSSSSSS           YYY
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT        PPP              SSSSSSSSSSSS           YYY
```

```
_$

Va
--
00
00
00
00
00
48
7F
7F
7F
7F
7F
7F
7F
7F
7F
7F
7F
7F
7F
7F
7F
7F
7F
```

```
                                              D 14
**FILE**ID**SATSSS50





  SSSSSSSS    AAAAAA    TTTTTTTTTT    SSSSSSSS    SSSSSSSS    SSSSSSSS  5555555555    000000
  SSSSSSSS    AAAAAA    TTTTTTTTTT    SSSSSSSS    SSSSSSSS    SSSSSSSS  5555555555    000000
SS          AA      AA      TT      SS          SS          SS          55          00      00
SS          AA      AA      TT      SS          SS          SS          55          00      00
SS          AA      AA      TT      SS          SS          SS          555555      00    0000
SS          AA      AA      TT      SS          SS          SS          555555      00    0000
  SSSSSS    AA      AA      TT        SSSSSS      SSSSSS      SSSSSS          55    00  00  00
  SSSSSS    AA      AA      TT        SSSSSS      SSSSSS      SSSSSS          55    00  00  00
        SS  AAAAAAAAAA      TT              SS          SS          SS          55  0000    00
        SS  AAAAAAAAAA      TT              SS          SS          SS          55  0000    00
        SS  AA      AA      TT              SS          SS          SS  55      55  00      00      ....
        SS  AA      AA      TT              SS          SS          SS  55      55  00      00      ....
SSSSSSSS    AA      AA      TT      SSSSSSSS    SSSSSSSS    SSSSSSSS      555555      000000        ....
SSSSSSSS    AA      AA      TT      SSSSSSSS    SSSSSSSS    SSSSSSSS      555555      000000        ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
0000          .TITLE  SATSSS50 SATS SYSTEM SERVICE TESTS  $ASCEFC (SUCC S.C.)
0000          .IDENT  'V04-000'
0000
0000    ;
0000    ;*************************************************************************
0000    ;*                                                                       *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                              *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.               *
0000    ;*  ALL RIGHTS RESERVED.                                                 *
0000    ;*                                                                       *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY *
0000    ;*  TRANSFERRED.                                                         *
0000    ;*                                                                       *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT *
0000    ;*  CORPORATION.                                                         *
0000    ;*                                                                       *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.              *
0000    ;*                                                                       *
0000    ;*                                                                       *
0000    ;*************************************************************************
0000    ;
0000
0000    ;++
0000    ; FACILITY:     SYSTST (SATS SYSTEM SERVICE TESTS)
0000    ;
0000    ; ABSTRACT:
0000    ;
0000    ;               THIS MODULE CONTAINS SUBROUTINES WHICH, WHEN LINKED
0000    ; WITH SUCCOMMON.OBJ, FORM TEST MODULE SATSSS50 TO TEST SUCCESSFUL
0000    ; OPERATION OF THE $ASCEFC SYSTEM SERVICE. THE SERVICE IS INVOKED
0000    ; UNDER VARIOUS INPUT CONDITIONS WITH VARYING INPUT PARAMETERS. ONLY
0000    ; SUCCESSFUL STATUS CODES ARE EXPECTED IN THIS TEST MODULE.  CORRECT
0000    ; OPERATION OF THE SERVICE FOR EACH OF ITS ISSUANCES IS VERIFIED BY
0000    ; CHECKING FOR AN SS$_NORMAL STATUS CODE, EXPECTED RETURN ARGUMENTS
0000    ; AND EXPECTED FUNCTIONALITY PERFORMED.
0000    ;
0000    ; ENVIRONMENT:  USER MODE IMAGE; NEEDS CMKRNL PRIVILEGE,
0000    ;               DYNAMICALLY ACQUIRES OTHER PRIVILEGES, AS NEEDED.
0000    ;
0000    ; AUTHOR: THOMAS L. CAFARELLA,          CREATION DATE: DEC, 1977
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;               VERSION 1.5 : 25-MAY-79
0000    ;
0000    ; 01  LDJ  10/11/79     Fixed bug caused by DIB$K_LENGTH change ACG052.RNO mem
0000    ;
0000    ;--
```

```
                    0000            .SBTTL  DECLARATIONS
                    0000    ;
                    0000    ; INCLUDE FILES:
                    0000    ;
                    0000            $PRVDEF                         ; PRIVILEGE BIT DEFINITIONS
                    0000            $PHDDEF                         ; PROCESS HEADER OFFSETS
                    0000            $PQLDEF                         ; PROCESS QUOTA CODES
                    0000            $DIBDEF                         ; DEVICE INFO BLOCK OFFSETS
                    0000    ;
                    0000    ; MACROS:
                    0000    ;
                    0000    ;
                    0000    ; EQUATED SYMBOLS:
                    0000    ;
                    0000    ;
                    0000    ; BIT NUMBERS FOR FLAGS CONTAINED IN "FLAGS" BYTE:
                    0000    ;
          00000000  0000    FLG_V_CLAOTHEV  = 0                     ; ASCEFC ISSUED FOR CLUS A, OTHER E.F. GROUP
          00000001  0000    FLG_V_CLAPROC   = 1                     ; ASCEFC ISSUED FOR CLUSTER A, ...
                    0000                                            ; ... EVENT FLAG GROUP 2, CREATED PROC
          00000002  0000    FLG_V_MKFORMED  = 2                     ; CLUSTER A MASK HAS BEEN ...
                    0000                                            ; ... FORMED FOR THIS TEST CASE
                    0000    ;
                    0000    ; OWN STORAGE:
                    0000    ;
```

```
00000000               .PSECT  RODATA,RD,NOWRT,NOEXE,LCNG
    0000   TEST_MOD_NAME:: STRING  C,<SATSSS50>     ; TEST MODULE NAME
    0009   TEST_MOD_NAME_D: STRING I,<SATSSS50>     ; TEST MODULE NAME DESCRIPTOR
    0019   MSG1_INP_CTL:   STRING  I,<  SSASC!4ZW: CONDITIONS:>
    0039                                            ; FAO CTL STRING FOR MSG1 IN SUCCOMMON.MAR
    0039   MSG3_ERR_CTL::  STRING  I,< *SSASC!4ZW: !AS>
    0051                                            ; FAO CTL STRING FOR MSG3 IN SUCCOMMON.MAR
    0051   CREPRN:         STRING  I,<SATSSS50_CRE> ; CREATED PROCESS NAME
    0065   CLUS_NAME_A:    STRING  I,<SATSSS50_CLA> ; CLUSTER A NAME
    0079   CLUS_NAME_B:    STRING  I,<SATSSS50_CLB> ; CLUSTER B NAME
    008D   IMAGNAM:        STRING  I,<SYSTST$RES:SATSUT04.EXE> ; IMAGE NAME FOR CREATED PROC
    00AC   QUOTALIST:      $QUOTA  CPULM,0          ; INFINITE CPU
    00B1                   $QUOTA  BYTLM,512        ; BYTE LIMIT FOR BUFFERED I/O
    00B6                   $QUOTA  FILLM,2          ; OPEN FILE COUNT LIMIT
    00BB                   $QUOTA  PGFLQUOTA,10     ; PAGING FILE QUOTA
    00C0                   $QUOTA  PRCLM,2          ; SUBPROCESS QUOTA
    00C5                   $QUOTA  TQELM,3          ; TIMER QUEUE ENTRY QUOTA
    00CA                   $QUOTA  LISTEND          ; DEFINES END OF LIST
```

```
   00000000                .PSECT  RWDATA,RD,WRT,NOEXE,LONG
00000008  0000   PRIVMASK:        .BLKQ    1                    ; ADDR OF PRIVILEGE MASK (IN PHD)
0000000C  0008   MBXCHAN:         .BLKL    1                    ; CHAN NO. FOR MAILBOX FOR CREATED PROCESS
          000C   MBXCHANINFO:                                   ; CHANNEL INFO RETURNED BY GETCHN
00000074  000C                    .LONG    DIB$K_LENGTH
00000014' 0010                    .ADDRESS .+4
00000088  0014                    .BLKB    DIB$K_LENGTH
0000008C  0088   MBXUNIT:         .BLKL    1                    ; SAVE AREA FOR MAILBOX UNIT NUMBER
          008C   MBXBUFF:         STRING   0,120                ; MAILBOX BUFFER FOR CREATED PROCESS
0000010D  010C   ASCTOT:          .BLKB    1                    ; NO. OF ASCEFC'S (REF COUNT) FOR CLUSTER A
00000111  010D   OTHER_EFN:       .BLKL    1                    ; SAVE AREA FOR "OTHER THAN SUBJECT" EFN
00000115  0111   CLUS_MASK:       .BLKL    1                    ; CLUSTER MASK  ; USED TO SET CLUSTER A
00000119  0115   CLUS_STATE:      .BLKL    1                    ; STATE OF CLUSTER A
      00  0119   FLAGS:           .BYTE    0                    ; GEN. PURP. FLAGS; BIT DEFINITIONS ABOVE
```

```
                     011A        .SBTTL  CONDITION TABLES
                     011A    ;
                     011A    ;   ***** CONDITION TABLES FOR ASCEFC SYSTEM SERVICE *****
                     011A    ;
                     011A        COND    1,LONG,<PERM>,-
                     011A                  <PERMANENT>,-
                     011A                  <TEMPORARY>,-
                     011A
           00000001  013C                  .LONG         1       ; PERMANENT CLUSTER
           00000000  0140                  .LONG         0       ; TEMPORARY CLUSTER
                     0144    ;
                     0144        COND    2,NOTARG,<PRE-EXISTING ASSOCIATION>,-
                     0144                <EVENT FLAG GROUP NOT ALREADY ASSOCIATED>,-
                     0144                <EVENT FLAG GROUP ALREADY ASSOCIATED TO SUBJECT CLUSTER>,-
                     0144                <EVENT FLAG GROUP ALREADY ASSOCIATED TO NON-SUBJECT CLUSTER>,-
                     0144
                     0204        COND    3,NOTARG,<REFERENCE COUNT FOR SUBJECT CLUSTER>,-
                     0204                <ZERO>,-
                     0204                <ONE>,-
                     0204                <GREATER THAN ONE (TWO)>,-
                     0204
           02 01 00  0255                  .BYTE         0,1,2
                     0258    ;
                     0258        COND    4,LONG,<EFN>,-
                     0258                <EVENT FLAGS 64-95 (EV FLAG GROUP 2)>,-
                     0258                <EVENT FLAGS 96-127 (EV FLAG GROUP 3)>,-
                     0258
           00000040  02AE                  .LONG         64      ; EVENT FLAG GROUP 2
           00000060  02B2                  .LONG         96      ; EVENT FLAG GROUP 3
                     02B6    ;
                     02B6        COND    5,NULL
                     02B7
                 00000000        .PSECT  SATSSS50,RD,WRT,EXE
```

```
                                0000          .SBTTL  TM_SETUP, TM_CLEANUP
                                0000    ;++
                                0000    ; FUNCTIONAL DESCRIPTION:
                                0000    ;
                                0000    ;               TM_SETUP AND TM_CLEANUP ARE CALLED TO PERFORM
                                0000    ; REQUIRED HOUSEKEEPING AT THE BEGINNING AND END, RESPECTIVELY, OF
                                0000    ; TEST MODULE EXECUTION.
                                0000    ;
                                0000    ; CALLING SEQUENCE:
                                0000    ;
                                0000    ;       BSBW TM_SETUP   BSBW TM_CLEANUP
                                0000    ;
                                0000    ; INPUT PARAMETERS:
                                0000    ;
                                0000    ;       NONE
                                0000    ;
                                0000    ; IMPLICIT INPUTS:
                                0000    ;
                                0000    ;       NONE
                                0000    ;
                                0000    ; OUTPUT PARAMETERS:
                                0000    ;
                                0000    ;       NONE
                                0000    ;
                                0000    ; IMPLICIT OUTPUTS:
                                0000    ;
                                0000    ;       TM_SETUP:  COND TABLE INDEX REGISTERS (R2,3,4,5,6) CLEARED;
                                0000    ;                  ALL PRIVILEGES ACQUIRED.
                                0000    ;
                                0000    ; COMPLETION CODES:
                                0000    ;
                                0000    ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
                                0000    ;
                                0000    ; SIDE EFFECTS:
                                0000    ;
                                0000    ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
                                0000    ;       (VIA RSB) IF ERROR ENCOUNTERED.
                                0000    ;
                                0000    ;--
                                0000
                                0000
                                0000
                                0000    TM_SETUP::
                      52   D4   0000            CLRL    R2                              ; INITIALIZE
                      53   D4   0002            CLRL    R3                              ; .. CONDITION
                      54   D4   0004            CLRL    R4                              ; .... TABLE
                      55   D4   0006            CLRL    R5                              ; ...... INDEX
                      56   D4   0008            CLRL    R6                              ; ........ REGISTERS
                   FFF3'   30   000A            BSBW    MOD_MSG_PRINT                   ; PRINT TEST MODULE BEGIN MSG
00000000'EF  00000000'EF   DE   000D            MOVAL   TEST_MOD_SUCC,TMD_ADDR          ; ASSUME END MSG WILL SHOW SUCCESS
    03   00  00000000'8F   F0   0018            INSV    #SUCCESS,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR SUCCESS
             00000000'EF        0020
                                0025            MODE    TO,5$,KRNL                      ; KERNEL MODE TO ACCESS PHD
         59  00000000'9F   D0   0048            MOVL    @#CTL$GL_PHD,R9                 ; GET PROCESS HEADER ADDRESS
      00000000'EF    69    DE   004F            MOVAL   PHD$Q_PRIVMSK(R9),PRIVMASK ; GET PRIV MASK ADDRESS
                                0056            MODE    FROM,5$ ; BACK TO USER MODE
                                0057            PRIV    ADD,ALL                         ; GET ALL PRIVILEGES
                                0077          $SETPRN_S TEST_MOD_NAME_D         ; SET PROCESS NAME
                                0084          SS_CHECK  NORMAL                  ; CHECK STATUS CODE RETURNED FROM SETPRN
                                00B2          $CREMBX_S CHAN=MBXCHAN, LOGNAM=CREPRN, - ; GET MAILBOX FOR PROCESS
                                00B2                    MAXMSG=#120, PROMSK=#0, BUFQUO=#240
                                00D7          SS_CHECK NORMAL                   ; CHECK NORMAL COMPLETION
                                0105          $GETCHN_S CHAN=MBXCHAN, -
                                0105                    PRIBUF=MBXCHANINFO      ; GET CHAN INFO (UNIT NUMBER)
                                011F          SS_CHECK NORMAL                   ; CHECK NORMAL COMPLETION
 00000088'EF   00000020'EF   3C 014D          MOVZWL  MBXCHANINFO+8+DIB$W_UNIT,MBXUNIT ; SAVE MAILBOX UNIT NUMBER
                             05 0158          RSB                               ; RETURN TO MAIN ROUTINE
                                0159  TM_CLEANUP::
                                0159          $DELMBX_S  MBXCHAN                ; DELETE TERMINATION MAILBOX
                      FE96'  30 0167          BSBW    MOD_MSG_PRINT             ; PRINT TEST MODULE END MSG
                             05 016A          RSB                               ; RETURN TO MAIN ROUTINE
```

```
     016B          .SBTTL  CONDITION SUBROUTINES - SETUP AND CLEANUP
     016B   ;++
     016B   ; FUNCTIONAL DESCRIPTION:
     016B   ;
     016B   ;               CONDX AND CONDX_CLEANUP ARE SUBROUTINES WHICH ARE EXECUTED
     016B   ; BEFORE AND AFTER THE VERIFY SUBROUTINE, RESPECTIVELY, WHENEVER A NEW
     016B   ; CONDITION X VALUE IS SELECTED (SEE FUNCTIONAL DESCRIPTION OF SUCCOMMON
     016B   ; ROUTINE IN SUCCOMMON.MAR). ANY SETUP FUNCTION PARTICULAR TO THE
     016B   ; CONDITION X TABLE IS INCLUDED IN THE CONDX SUBROUTINE AND CLEANED
     016B   ; UP, IF NECESSARY, IN THE CONDX_CLEANUP SUBROUTINE. THIS INCLUDES,
     016B   ; ESPECIALLY, CODE TO DETECT CONFLICTS AMONG CURRENT ENTRIES IN TWO
     016B   ; OR MORE CONDITION TABLES. IF A CONFLICT IS DETECTED, A NON-ZERO
     016B   ; VALUE IS STORED INTO CONFLICT, WHICH CAUSES THE CALLING ROUTINE
     016B   ; (SUCCOMMON) TO SKIP THE CURRENT ENTRY IN THE CONDITION X TABLE.
     016B   ;
     016B   ; CALLING SEQUENCE:
     016B   ;
     016B   ;       BSBW CONDX   BSBW CONDX_CLEANUP
     016B   ;         WHERE X = 1,2,3,4,5
     016B   ;
     016B   ; INPUT PARAMETERS:
     016B   ;
     016B   ;       CONFLICT = 0
     016B   ;
     016B   ; IMPLICIT INPUTS:
     016B   ;
     016B   ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
     016B   ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
     016B   ;
     016B   ; OUTPUT PARAMETERS:
     016B   ;
     016B   ;       CONFLICT SET TO NON-ZERO IF COND TABLE CONFLICT DETECTED.
     016B   ;
     016B   ; IMPLICIT OUTPUTS:
     016B   ;
     016B   ;       R2,3,4,5,6 PRESERVED
     016B   ;
     016B   ; COMPLETION CODES:
     016B   ;
     016B   ;       NONE
     016B   ;
     016B   ; SIDE EFFECTS:
     016B   ;
     016B   ;       NONE
     016B   ;
     016B   ;--
     016B
     016B
     016B
     016B   COND1::
 05  016B           RSB                             ; RETURN TO MAIN ROUTINE
     016C   COND1_CLEANUP::
 05  016C           RSB                             ; RETURN TO MAIN ROUTINE
     016D   COND2::
 05  016D           RSB                             ; RETURN TO MAIN ROUTINE
     016E   COND2_CLEANUP::
 05  016E           RSB                             ; RETURN TO MAIN ROUTINE
```

```
                                 016F   COND3::
          00000255'EF44     95   016F             TSTB    COND3 E[R4]             ; IS REFERENCE COUNT ZERO ?
                     10     12   0176             BNEQU   COND3X                  ; NO -- ALL IS OK, JUST EXIT
               01    53     91   0178             CMPB    R3,#1                   ; YES -- DOES COND 2 SPECIFY SAME CLUSTER ?
                     0B     12   017B             BNEQU   COND3X                  ; NO -- ALL IS OK, JUST EXIT
   00000000'EF   00000000'EF 90  017D             MOVB    ONES,CONFLICT           ; YES -- INDICATE CONFLICT
                                 0188   COND3X:
                            05   0188             RSB                             ; RETURN TO MAIN ROUTINE
                                 0189   COND3_CLEANUP::
                            05   0189             RSB                             ; RETURN TO MAIN ROUTINE
                                 018A   COND4::
                            05   018A             RSB                             ; RETURN TO MAIN ROUTINE
                                 018B   COND4_CLEANUP::
                            05   018B             RSB                             ; RETURN TO MAIN ROUTINE
                                 018C   COND5::
                            05   018C             RSB                             ; RETURN TO MAIN ROUTINE
                                 018D   COND5_CLEANUP::
                            05   018D             RSB                             ; RETURN TO MAIN ROUTINE
```

```
                                  018E        .SBTTL  FORM_CONDS
                                  018E  ;++
                                  018E  ; FUNCTIONAL DESCRIPTION:
                                  018E  ;
                                  018E  ;             FORM_CONDS FORMATS AND PRINTS INFORMATION ABOUT
                                  018E  ;  THE CURRENT ELEMENT IN EACH OF THE CONDITION TABLES.
                                  018E  ;
                                  018E  ; CALLING SEQUENCE:
                                  018E  ;
                                  018E  ;       BSBW FORM_CONDS
                                  018E  ;
                                  018E  ; INPUT PARAMETERS:
                                  018E  ;
                                  018E  ;       NONE
                                  018E  ;
                                  018E  ; IMPLICIT INPUTS:
                                  018E  ;
                                  018E  ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
                                  018E  ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
                                  018E  ;       FOR X = 1,2,3,4,5 :
                                  018E  ;               CONDX_T - TITLE TEXT FOR CONDX TABLE
                                  018E  ;               CONDX_TAB - ELEMENT TEXT FOR CONDX TABLE
                                  018E  ;               CONDX_C - CONTEXT OF THE CONDX TABLE
                                  018E  ;               CONDX_E - DATA ELEMENTS OF THE CONDX TABLE
                                  018E  ;
                                  018E  ; OUTPUT PARAMETERS:
                                  018E  ;
                                  018E  ;       NONE
                                  018E  ;
                                  018E  ; IMPLICIT OUTPUTS:
                                  018E  ;
                                  018E  ;       NONE
                                  018E  ;
                                  018E  ; COMPLETION CODES:
                                  018E  ;
                                  018E  ;       NONE
                                  018E  ;
                                  018E  ; SIDE EFFECTS:
                                  018E  ;
                                  018E  ;       NONE
                                  018E  ;
                                  018E  ;--
                                  018E
                                  018E
                                  018E
                                  018E  FORM_CONDS::
                                  018E          $FAO_S  MSG1_INP_CTL,FAO_LEN,FAO_DESC,TESTNUM
                                  01AD                                          ; FORMAT CONDITIONS HEADER MSG
                  FE50'     30    01AD          BSBW    OUTPUT_MSG              ; ... AND PRINT IT
             14      04     91    01B0          CMPB    #COND1_C,#NULL          ; IS CONDITION 1 NULL ?
                     03     12    01B3          BNEQU   10$                     ; NO -- CONTINUE
                   00D7     31    01B5          BRW     FORM_CONDSX             ; YES -- SUBROUTINE IS FINISHED
                                  01B8  10$:
 00000000'EF   0000011A'EF  DE    01B8          MOVAL   COND1_T,MSG_A           ; SAVE ADDRESS OF CONDITION 1 TITLE FOR FAO
00000000'EF   00000120'EF42 D0    01C3          MOVL    COND1_TAB[R2],MSG_B     ; SAVE ADDR OF COND 1 CURR TEXT ELT FOR FAO
           00000000'EF   04 90    01CF          MOVB    #CONDT_C,MSG_CTXT       ; SAVE CONDITION 1 CONTEXT FOR FAO
                                  01D6          MOV_VAL COND1_C,CONDT_E[R2],MSG_DATA1 ; GIVE COND 1 DATA VALUE TO FAO
```

```
                      FE1B'  30  01E2                   BSBW     WRITE_MSG2                    ; FORMAT AND WRITE CONDITION 1 MSG
                  14    00   91  01E5                   CMPB     #COND2_C,#NULL                ; IS CONDITION 2 NULL ?
                        03   12  01E8                   BNEQU    20$                           ; NO -- CONTINUE
                      00A2   31  01EA                   BRW      FORM_CONDSX                   ; YES -- SUBROUTINE IS FINISHED
                                 01ED       20$:
  00000000'EF    00000144'EF DE  01ED                   MOVAL    COND2_T,MSG_A                 ; SAVE ADDRESS OF CONDITION 2 TITLE FOR FAO
00000000'EF    0000015E'EF43 D0  01F8                   MOVL     COND2_TAB[R3],MSG_B           ; SAVE ADDR OF COND 2 CURR TEXT ELT FOR FAO
              00000000'EF    00  90  0204               MOVB     #COND2_C,MSG_CTXT             ; SAVE CONDITION 2 CONTEXT FOR FAO
                                 020B                   MOV_VAL  COND2_C,COND2_E[R3],MSG_DATA1 ; GIVE COND 2 DATA VALUE TO FAO
                      FDF2'  30  020B                   BSBW     WRITE_MSG2                    ; FORMAT AND WRITE CONDITION 2 MSG
                  14    00   91  020E                   CMPB     #COND3_C,#NULL                ; IS CONDITION 3 NULL ?
                        03   12  0211                   BNEQU    30$                           ; NO -- CONTINUE
                      0079   31  0213                   BRW      FORM_CONDSX                   ; YES -- SUBROUTINE IS FINISHED
                                 0216       30$:
  00000000'EF    00000204'EF DE  0216                   MOVAL    COND3_T,MSG_A                 ; SAVE ADDRESS OF CONDITION 3 TITLE FOR FAO
00000000'EF    00000229'EF44 D0  0221                   MOVL     COND3_TAB[R4],MSG_B           ; SAVE ADDR OF COND 3 CURR TEXT ELT FOR FAO
              00000000'EF    00  90  022D               MOVB     #COND3_C,MSG_CTXT             ; SAVE CONDITION 3 CONTEXT FOR FAO
                                 0234                   MOV_VAL  COND3_C,COND3_E[R4],MSG_DATA1 ; GIVE COND 3 DATA VALUE TO FAO
                      FDC9'  30  0234                   BSBW     WRITE_MSG2                    ; FORMAT AND WRITE CONDITION 3 MSG
                  14    04   91  0237                   CMPB     #COND4_C,#NULL                ; IS CONDITION 4 NULL ?
                        53   13  023A                   BEQLU    FORM_CONDSX                   ; YES -- SUBROUTINE IS FINISHED
  00000000'EF    00000258'EF DE  023C                   MOVAL    COND4_T,MSG_A                 ; SAVE ADDRESS OF CONDITION 4 TITLE FOR FAO
00000000'EF    0000025D'EF45 D0  0247                   MOVL     COND4_TAB[R5],MSG_B           ; SAVE ADDR OF COND 4 CURR TEXT ELT FOR FAO
              00000000'EF    04  90  0253               MOVB     #COND4_C,MSG_CTXT             ; SAVE CONDITION 4 CONTEXT FOR FAO
                                 025A                   MOV_VAL  COND4_C,COND4_E[R5],MSG_DATA1 ; GIVE COND 4 DATA VALUE TO FAO
                      FD97'  30  0266                   BSBW     WRITE_MSG2                    ; FORMAT AND WRITE CONDITION 4 MSG
                  14    14   91  0269                   CMPB     #COND5_C,#NULL                ; IS CONDITION 5 NULL ?
                        21   13  026C                   BEQLU    FORM_CONDSX                   ; YES -- SUBROUTINE IS FINISHED
  00000000'EF    000002B6'EF DE  026E                   MOVAL    COND5_T,MSG_A                 ; SAVE ADDRESS OF CONDITION 5 TITLE FOR FAO
00000000'EF    000002B6'EF46 D0  0279                   MOVL     COND5_TAB[R6],MSG_B           ; SAVE ADDR OF COND 5 CURR TEXT ELT FOR FAO
              00000000'EF    14  90  0285               MOVB     #COND5_C,MSG_CTXT             ; SAVE CONDITION 5 CONTEXT FOR FAO
                                 028C                   MOV_VAL  COND5_C,COND5_E[R6],MSG_DATA1 ; GIVE COND 5 DATA VALUE TO FAO
                      FD71'  30  028C                   BSBW     WRITE_MSG2                    ; FORMAT AND WRITE CONDITION 5 MSG
                                 028F       FORM_CONDSX:
                             05  028F                   RSB                                    ; RETURN TO CALLER
```

```
0290            .SBTTL  VERIFY
0290    ;++
0290    ; FUNCTIONAL DESCRIPTION:
0290    ;
0290    ;               VERIFY IS CALLED ONCE FOR EACH COMBINATION OF CONDITION
0290    ; TABLE VALUES (AS DETERMINED BY THE INDEX REGISTERS R2,3,4,5,6 FOR
0290    ; COND TABLES 1,2,3,4,5, RESPECTIVELY). VERIFY ESTABLISHES THE CONDITIONS
0290    ; SPECIFIED BY THE COND TABLES AND ISSUES THE SUBJECT SYSTEM SERVICE
0290    ; ($ASCEFC). THEN, THE SUCCESSFUL OPERATION OF THE SERVICE IS VERIFIED
0290    ; BY EXAMINING THE STATUS CODE RETURNED, THE VALUES FOR RETURN ARGUMENTS
0290    ; AND THE FUNCTIONALITY PERFORMED. THE EXAMINATIONS TAKE THE FORM OF
0290    ; COMPARISONS AGAINST EXPECTED VALUES. ANY FAILING COMPARISON CAUSES AN
0290    ; ERR_EXIT MACRO TO BE EXECUTED (EITHER DIRECTLY, OR INDIRECTLY,
0290    ; THROUGH THE SS_CHECK MACRO); ERR_EXIT SETS EFLAG TO NON-ZERO,
0290    ; PRINTS ERROR MESSAGES AND CAUSES AN IMMEDIATE RSB TO CALLER.
0290    ; WHEN ERR_EXIT IS EXECUTED, FURTHER CALLS TO VERIFY ARE SUPPRESSED,
0290    ; AND, AFTER EXECUTING CLEANUP SUBROUTINES, THE IMAGE EXITS.
0290    ;
0290    ; CALLING SEQUENCE:
0290    ;
0290    ;       BSBW VERIFY
0290    ;
0290    ; INPUT PARAMETERS:
0290    ;
0290    ;       NONE
0290    ;
0290    ; IMPLICIT INPUTS:
0290    ;
0290    ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
0290    ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
0290    ;       FOR X = 1,2,3,4,5 :
0290    ;               CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
0290    ;                 TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
0290    ;                 ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
0290    ;                 FOR CONDX_E.
0290    ;
0290    ; OUTPUT PARAMETERS:
0290    ;
0290    ;       NONE
0290    ;
0290    ; IMPLICIT OUTPUTS:
0290    ;
0290    ;       VERIFY HAS NO OUTPUT. SINCE ITS PURPOSE IS TO TEST FOR ERRORS,
0290    ;       IT MERELY RETURNS TO CALLER NORMALLY AFTER THE TESTS, PROVIDING
0290    ;       ALL WERE SUCCESSFUL; IF AN ERROR IS DISCOVERED, RETURN IS VIA
0290    ;       AN ERR_EXIT OR SS_CHECK MACRO, BOTH OF WHICH DOCUMENT DETECTED
0290    ;       ERRORS.
0290    ;
0290    ; COMPLETION CODES:
0290    ;
0290    ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
0290    ;
0290    ; SIDE EFFECTS:
0290    ;
0290    ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
0290    ;       (VIA RSB) IF ERROR ENCOUNTERED.
0290    ;
```

```
                             0290   447 ;--
                             0290   448
                             0290   449
                             0290   450
                             0290   451 VERIFY::
          00000000'EF    95  0290   452         TSTB    CFLAG                   ; SHOULD CONDITIONS BE PRINTED ?
                   03    13  0296   453         BEQL    5$                      ; NO -- CONTINUE
                 FEF3    30  0298   454         BSBW    FORM_CONDS              ; YES -- FMT & PRINT ALL CONDS FOR THIS T.C.
                             029B   455 5$:
          00000119'EF    94  029B   456         CLRB    FLAGS                   ; RE-INIT ALL FLAGS FOR NEXT TEST CASE
          0000010C'EF    94  02A1   457         CLRB    ASCTOT                  ; CLEAR REFERENCE CNT FOR (SUBJECT) CLUST A
                   53    D5  02A7   458         TSTL    R3                      ; EV FLAG GROUP HAVE PRIOR ASSOCIATION ?
                   7C    13  02A9   459         BEQL    11$                     ; NO -- CONTINUE
             53    01    D1  02AB   460         CMPL    #1,R3                   ; YES -- PRIOR ASSOCIATION WITH CLUSTER A ?
                   0F    12  02AE   461         BNEQ    7$                      ; NO -- MUST BE CLUSTER B
          0000010C'EF    96  02B0   462         INCB    ASCTOT                  ; YES -- INCR REF COUNT FOR CLUSTER A
     57   00000065'EF    DE  02B6   463         MOVAL   CLUS_NAME_A,R7          ; ... AND SET UP CLUSTER NAME FOR ASCEFC
                   07    11  02BD   464         BRB     9$                      ; GO ISSUE PRELIMINARY SERVICE
                             02BF   465 7$:
     57   00000079'EF    DE  02BF   466         MOVAL   CLUS_NAME_B,R7          ; SET UP CLUSTER NAME FOR ASCEFC
                             02C6   467 9$:
   5A   000002AE'EF45    D0  02C6   468         MOVL    EFN[R5],R10             ; EFN MUST BE IN R10 FOR LATER CALL
                             02CE   469         $ASCEFC_S EFN=R10, NAME=(R7), PERM=PERM[R2]
                             02E2   470         SS_CHECK NORMAL                 ; CHECK SERVICE COMPLETION
     57   00000065'8F    D1  0310   471         CMPL    #CLUS_NAME_A,R7         ; DID WE ASSOCIATE CLUSTER A ?
                   0E    12  0317   472         BNEQU   11$                     ; NO -- SKIP BUILDING OF CLUSTER A
                 067A    30  0319   473         BSBW    BUILD_CLUST             ; BUILD CLUSTER A
          00000000'EF    95  031C   474         TSTB    EFLAG                   ; IS AN ERROR BEING PROCESSED ?
                   03    13  0322   475         BEQL    11$                     ; NO -- CONTINUE
                 0625    31  0324   476         BRW     VERIFYX                 ; YES -- RETURN IMMEDIATELY
                             0327   477 11$:
00000255'EF44  0000010C'EF   83  0327   478         SUBB3   ASCTOT,COND3_E[R4],R7   ; CALC. NO. OF ASCEFC'S TO BE ISSUED
                   57        0333
                   57    95  0334   479         TSTB    R7                      ; ANY ASCEFC'S TO ISSUE ?
                   03    12  0336   480         BNEQU   14$                     ; YES -- CONTINUE
                 00E7    31  0338   481         BRW     25$                     ; NO -- GO ISSUE SUBJECT ASCEFC
                             033B   482 14$:
                   58    D4  033B   483         CLRL    R8                      ; ASSUME SECOND COND 4 ELEMENT
                   55    D5  033D   484         TSTL    R5                      ; FIRST COND 4 ELEMENT ?
                   02    12  033F   485         BNEQU   16$                     ; NO -- IT'S SECOND COND 4 ELEMENT
                   58    D6  0341   486         INCL    R8                      ; YES -- USE R8 AS INDEX TO 2ND ELEMENT
                             0343   487 16$:
   5A   000002AE'EF48    D0  0343   488         MOVL    EFN[R8],R10             ; GET EFN OF ''OTHER'' EV FLAG GROUP
     0000010D'EF    5A   D0  034B   489         MOVL    R10,OTHER_EFN           ; SAVE EFN OF ''OTHER'' GROUP
   00 00000119'EF   00   E2  0352   490         BBSS    #FLG_V_CLAOTHEV,FLAGS,.+1 ; INDICATE A LATER $DACEFC IS NEEDED
                             035A   491         $ASCEFC_S EFN=R10, NAME=CLUS_NAME_A, PERM=PERM[R2]
                             0372   492                                         ; ASSOC. ''OTHER'' EV FLAG GROUP WITH CLUST A
                             0372   493         SS_CHECK NORMAL                 ; CHECK FOR NORMAL COMPLETION
                 05F3    30  03A0   494         BSBW    BUILD_CLUST             ; BUILD CLUSTER A
          00000000'EF    95  03A3   495         TSTB    EFLAG                   ; IS AN ERROR BEING PROCESSED ?
                   03    13  03A9   496         BEQL    20$                     ; NO -- CONTINUE
                 059E    31  03AB   497         BRW     VERIFYX                 ; YES -- RETURN IMMEDIATELY
                             03AE   498 20$:
             57    02    91  03AE   499         CMPB    #2,R7                   ; MUST WE DO ANOTHER ASCEFC ?
                   6F    12  03B1   500         BNEQU   25$                     ; NO -- GO ISSUE SUBJECT ASCEFC
   00 00000119'EF   01   E2  03B3   501         BBSS    #FLG_V_CLAPROC,FLAGS,.+1 ; INDICATE A LATER $DACEFC IS NEEDED
                             03BB   502         $CREPRC_S PRCNAM=CREPRN, IMAGE=IMAGNAM, -
```

```
                                  03BB   503                  MBXUNT=MBXUNIT, QUOTA=QUOTALIST
                                  03ED   504                                                  ; ISSUE ASCEFC IN A CREATED PROCESS
                                  03ED   505          SS_CHECK NORMAL                         ; CHECK COMPLETION OF CREPRC
                                  041B   506          $HIBER_S                                ; SLEEP UNTIL CREATED PROCESS DOES $ASCEFC
                                  0422   507 25$:
                                  0422   508 ;
                                  0422   509 ; ****** SYSTEM SERVICE CALL WHICH IS THE SUBJECT OF THIS TEST CASE ******
                                  0422   510 ;
                                  0422   511          $ASCEFC_S EFN=EFN[R5], -
                                  0422   512                  NAME=CLUS_NAME_A, -
                                  0422   513                  PERM=PERM[R2]
         00000000'8F    50    D1  043F   514          CMPL    R0,#SS$_NORMAL                  ; CODE RECEIVED = CODE EXPECTED ?
                        61    13  0446   515          BEQLU   30$                             ; YES -- CONTINUE
00000000'EF    00000000'8F    D0  0448   516          MOVL    #SS$_NORMAL,EXPV                ; LOAD UP EXPECTED AND ...
         00000000'EF    50    D0  0453   517          MOVL    R0,RECV                         ; ... RECEIVED VALUES, THEN EXIT
                                  045A   518          ERR_EXIT LONG,<INCORRECT STATUS CODE RETURNED FROM ASCEFC>
                                  04A9   519 30$:
                                  04A9   520 ;
                                  04A9   521 ;  ENSURE THAT SETEF'S CAN BE PROPERLY ISSUED ON CLUSTER A
                                  04A9   522 ;    BY SETTING THE TWO HI-ORDER FLAGS OF THE CLUSTER (I.E.,
                                  04A9   523 ;    94-95 OR 126-127).
                                  04A9   524 ;
       57   000002AE'EF45     D0  04A9   525          MOVL    EFN[R5],R7                      ; GET FIRST EVENT FLAG OF CLUSTER INTO REG
                   57   1E    C0  04B1   526          ADDL2   #30,R7                          ; COMPUTE 2ND-TO-HIGHEST EV FLAG NO.
                                  04B4   527          $SETEF_S  EFN=R7                        ; ATTEMPT TO SET FLAG IN CLUSTER A
                                  04BD   528          SS_CHECK WASCLR                         ; FLAG SHOULD HAVE BEEN CLEAR (FROM ASCEFC)
                        57    D6  04EB   529          INCL    R7                              ; POINT REG 7 TO HIGHEST EV FLAG IN CLUSTER
                                  04ED   530          $SETEF_S  EFN=R7                        ; SET ANOTHER FLAG FOR GOOD MEASURE
                                  04F6   531          SS_CHECK WASCLR                         ; CHECK FOR PRIOR CLEAR CONDITION
                                  0524   532 ;
                                  0524   533 ;   SET UP REG 7 TO CONTAIN THE MASK OF EXPECTED EVENT FLAG SETTINGS
                                  0524   534 ;
    04 00000119'EF      02    E0  0524   535          BBS     #FLG_V_MKFORMED,FLAGS,40$ ; BRANCH IF CLUS_MASK FORMED
                        57    D4  052C   536          CLRL    R7                              ; SUBJECT ASCEFC GETS NEW CLUSTER; 0 MASK
                        07    11  052E   537          BRB     45$                             ; GO SET 2 MORE MASK BITS (FOR SETEF'S ABOVE
                                  0530   538 40$:
       57    00000111'EF      D0  0530   539          MOVL    CLUS_MASK,R7                    ; USE EXISTING CLUS_MASK
                                  0537   540 45$:
       57    02    1E   03    F0  0537   541          INSV    #^B11,#30,#2,R7                 ; TURN ON 2 HI-ORDER MASK BITS FOR SETEF'S
                                  053C   542          $READEF_S EFN=EFN[R5], STATE=CLUS_STATE
                                  0550   543                                                  ; READ CURRENT STATE OF CLUSTER A
                     2E 50    E8  0550   544          BLBS    R0,50$                          ; CONTINUE IF NORMAL COMPLETION
                                  0553   545          SS_CHECK NORMAL                         ; USE SS_CHECK TO TERMINATE TEST MODULE
                                  0581   546 50$:
       57    00000115'EF      D1  0581   547          CMPL    CLUS_STATE,R7                   ; IS CLUSTER A STATE = THAT EXPECTED ?
                        6B    13  0588   548          BEQLU   55$                             ; YES -- CONTINUE WITH VERIFICATION
         00000000'EF    57    D0  058A   549          MOVL    R7,EXPV                         ; NO -- LOAD EXPECTED AND ...
00000000'EF    00000115'EF    D0  0591   550          MOVL    CLUS_STATE,RECV                 ; ... RECEIVED VALUES, THEN EXIT
                                  059C   551          ERR_EXIT LONG,<PRE-EXISTING CLUSTER STATE NOT OBTAINED AFTER ASCEFC>
                                  05F5   552 55$:
                                  05F5   553      ;
                                  05F5   554      ; TO VERIFY THE ASCEFC REFERENCE COUNT, THE FOLLOWING CODE
                                  05F5   555      ;   ISSUES A DACEFC FOR EACH ASCEFC ISSUED BY THIS TEST CASE.
                                  05F5   556      ;
                                  05F5   557          $DACEFC_S EFN=EFN[R5]                   ; DISASSOCIATE SUBJECT ASCEFC
                                  0603   558          SS_CHECK NORMAL                         ; MAKE SURE IT COMPLETED OK
                                  0631   559 60$:
```

```
        0A 00000119'EF   01   E4  0631   560          BBSC    #FLG_V_CLAPROC,FLAGS,62$
                                  0639   561                                          ; HAVE CREATED PROCESS ISSUE DACEFC IF NEC.
        05 00000119'EF   00   E4  0639   562          BBSC    #FLG_V_CLAOTHEV,FLAGS,63$ ; ISSUE ANOTHER DACEFC IF NEC.
                         06   11  0641   563          BRB     64$                     ; ALL FLAGS CLEAR; REF COUNT 0
                                  0643   564 62$:
                       0179   31  0643   565          BRW     80$                     ; NEED A WORD'S WORTH OF BRANCH
                                  0646   566 63$:
                       020D   31  0646   567          BRW     85$                     ; NEED A WORD'S WORTH OF BRANCH
                                  0649   568 64$:
                                  0649   569 ;
                                  0649   570 ;    AT THIS POINT, ASCEFC REFERENCE COUNT SHOULD BE 0.
                                  0649   571 ;    ISSUE ANOTHER ASCEFC TO CHECK PERM/TEMP SETTING OF CLUSTER A
                                  0649   572 ;
                                  0649   573          $ASCEFC_S EFN=EFN[R5], NAME=CLUS_NAME_A ; ONE MORE ASSOCIATE
                                  0661   574          SS_CHECK NORMAL                 ; CHECK IT
                                  068F   575          $READEF_S EFN=EFN[R5], STATE=CLUS_STATE ; READ CLUSTER A
                      2E 50   E8  06A3   576          BLBS    R0,65$                  ; CONTINUE IF NORMAL COMPLETION
                                  06A6   577          SS_CHECK NORMAL                 ; USE SS_CHECK TO TERMINATE TEST MODULE
                                  06D4   578 65$:
              0000013C'EF42   D5  06D4   579          TSTL    PERM[R2]                ; IS THIS A PERMANENT CLUSTER ?
                         02   12  06DB   580          BNEQU   70$                     ; YES -- KEEP EXPECTED STATE VALUE FRM ABOVE
                         57   D4  06DD   581          CLRL    R7                      ; NO -- EXPECT A ZERO CLUSTER
                                  06DF   582 70$:
          57   00000115'EF   D1   06DF   583          CMPL    CLUS_STATE,R7           ; CLUSTER A STATE = THAT EXPECTED ?
                         5D   13  06E6   584          BEQLU   71$                     ; YES -- GO FINISH UP
             00000000'EF   57   D0  06E8   585          MOVL    R7,EXPV                 ; NO -- LOAD EXPECTED AND ...
 00000000'EF   00000115'EF   D0   06EF   586          MOVL    CLUS_STATE,RECV         ; ... RECEIVED VALUES, THEN EXIT
                                  06FA   587          ERR_EXIT LONG,<INCORRECT CLUSTER STATE AFTER DACEFC'S>
                                  0745   588 71$:
                                  0745   589          $DLCEFC_S NAME=CLUS_NAME_A      ; CLEAR PERM INDICATOR IF PRESENT
                                  0752   590          SS_CHECK NORMAL                 ; EXPECT NORMAL COMPLETION
                                  0780   591          $DACEFC_S EFN=EFN[R5]           ; ... AND DISASSOCIATE
                                  078E   592          SS_CHECK NORMAL                 ; ...........
                       018D   31  07BC   593          BRW     VERIFYX                 ; THIS TEST CASE IS COMPLETE
```

```
                                 07BF   595 80$:
                                 07BF   596          $WAKE_S PRCNAM=CREPRN           ; WAKE CREATED PROCESS TO GET DACEFC ISSUED
                                 07CE   597          SS_CHECK NORMAL                 ; CHECK FOR NORMAL STATUS CODE
                                 07FC   598          $QIOW_S CHAN=MBXCHAN, FUNC=#IO$_READVBLK, -
                                 07FC   599                  P1=MBXBUFF+8, P2=MBXBUFF
                                 0825   600                                          ; ... AND  WAIT FOR IT TO SEND MAIL
                                 0825   601          SS_CHECK NORMAL                 ; CHECK FOR NORMAL STATUS CODE
                     FDDB   31   0853   602          BRW     60$                     ; GO CHECK FOR MORE DACEFC'S
                                 0856   603 85$:
                                 0856   604          $READEF_S EFN=OTHER_EFN, STATE=CLUS_STATE
                                 0869   605                                          ; READ & CHECK CLUSTER BEFORE DACEFC
                   2E 50    E8   0869   606          BLBS    R0,86$                  ; CONTINUE IF NORMAL COMPLETION
                                 086C   607          SS_CHECK NORMAL                 ; USE SS_CHECK TO TERMINATE TEST MODULE
                                 089A   608 86$:
          57   00000115'EF  D1   089A   609          CMPL    CLUS_STATE,R7           ; CLUSTER A STATE = THAT EXPECTED ?
                        6B  13   08A1   610          BEQLU   87$                     ; YES -- GO DISASSOCIATE
            00000000'EF 57  D0   08A3   611          MOVL    R7,EXPV                 ; NO -- LOAD EXPECTED AND ...
  00000000'EF  00000115'EF  D0   08AA   612          MOVL    CLUS_STATE,RECV         ; ... RECEIVED VALUES, THEN EXIT
                                 08B5   613          ERR_EXIT LONG,<PRE-EXISTING CLUSTER STATE NOT OBTAINED AFTER DACEFC>
                                 090E   614 87$:
                                 090E   615          $DACEFC_S EFN=OTHER_EFN         ; DISASSOC "OTHER" EV FLAG GROUP FROM CLUS A
                                 091B   616          SS_CHECK NORMAL                 ; CHECK FOR NORMAL COMPLETION
                     FCE5   31   0949   617          BRW     60$                     ; GO CHECK FOR MORE FLAGS
                                 094C   618 VERIFYX:
                            05   094C   619          RSB                             ; RETURN TO CALLER
```

```
        094D            .SBTTL  VFY_CLEANUP
        094D    ;++
        094D    ; FUNCTIONAL DESCRIPTION:
        094D    ;
        094D    ;               VFY_CLEANUP EXECUTES SYSTEM SERVICES TO UNDO THE
        094D    ; EFFECT OF THOSE ISSUED IN THE VERIFY SUBROUTINE. VFY_CLEANUP MUST
        094D    ; ASSUME THAT VERIFY MAY NOT HAVE EXECUTED IN ITS ENTIRETY (IF AN
        094D    ; ERROR IS FOUND). ALSO, VFY_CLEANUP MAY ISSUE SS_CHECK OR ERR_EXIT
        094D    ; ONLY AFTER PERFORMING ALL OF ITS CLEANUP OPERATIONS; THIS IS REQUIRED
        094D    ; IN THE EVENT THAT VFY_CLEANUP IS CALLED DURING ERROR PROCESSING,
        094D    ; WHEN PERFORMING THE REQUIRED CLEANUP IS MORE IMPORTANT THAN
        094D    ; POSSIBLY DISCOVERING A SECOND ERROR.
        094D    ;
        094D    ; CALLING SEQUENCE:
        094D    ;
        094D    ;       BSBW VFY_CLEANUP
        094D    ;
        094D    ; INPUT PARAMETERS:
        094D    ;
        094D    ;       NONE
        094D    ;
        094D    ; IMPLICIT INPUTS:
        094D    ;
        094D    ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
        094D    ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
        094D    ;       FOR X = 1,2,3,4,5 :
        094D    ;               CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
        094D    ;                 TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
        094D    ;                 ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
        094D    ;                 FOR CONDX_E.
        094D    ;
        094D    ; OUTPUT PARAMETERS:
        094D    ;
        094D    ;       NONE
        094D    ;
        094D    ; IMPLICIT OUTPUTS:
        094D    ;
        094D    ;       NONE
        094D    ;
        094D    ; COMPLETION CODES:
        094D    ;
        094D    ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
        094D    ;
        094D    ; SIDE EFFECTS:
        094D    ;
        094D    ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
        094D    ;       (VIA RSB) IF ERROR ENCOUNTERED.
        094D    ;
        094D    ;--
        094D
        094D
        094D
        094D    VFY_CLEANUP::
        094D            $DLCEFC_S NAME=CLUS_NAME_A      ; CLEAR PERM INDICATORS IF PRESENT ...
        095A            $DLCEFC_S NAME=CLUS_NAME_B      ; ... FOR BOTH CLUSTERS
        0967            SS_CHECK NORMAL                 ; CHECK COMPLETION
     05 0995            RSB                             ; RETURN TO CALLER
```

```
                            0996          .SBTTL  BUILD_CLUST SUBROUTINE
                            0996     ;
                            0996     ; ***************************************************************
                            0996     ; *
                            0996     ; *      BUILD_CLUST SUBROUTINE
                            0996     ; *
                            0996     ; *      THIS SUBROUTINE CREATES A 32-BIT CLUSTER MASK BY
                            0996     ; *        CONCATENATING THE LOW-ORDER BYTES OF REGS R2-R5.
                            0996     ; *        IT THEN SETS CLUSTER A EQUAL TO THE MASK BY
                            0996     ; *        ISSUING THE PROPER COMBINATION OF 32 SETEF/CLREF'S.
                            0996     ; *
                            0996     ; *      INPUTS:
                            0996     ; *
                            0996     ; *              R2,R3,R4,R5   -  CONDITION TABLE INDEX VALUES
                            0996     ; *
                            0996     ; *              R10      -       ANY EFN IN CLUSTER A
                            0996     ; *
                            0996     ; *      OUTPUTS:
                            0996     ; *
                            0996     ; *              CLUS_MASK   -    LONGWORD CONTAINING THE CREATED
                            0996     ; *                                 CLUSTER MASK.
                            0996     ; *
                            0996     ; *              CLUSTER A   -    THE SUBJECT EVENT FLAG CLUSTER,
                            0996     ; *                                 UPDATED TO LOOK LIKE CLUS_MASK.
                            0996     ; *
                            0996     ; *              FLG_V_MKFORMED - BIT IN FLAGS BYTE IS SET, IND-
                            0996     ; *                                 ICATING CLUS_MASK IS FORMED.
                            0996     ; *
                            0996     ; *      VOLATILE REGISTERS:
                            0996     ; *
                            0996     ; *              R0, R1, R8, R9
                            0996     ; *
                            0996     ; ***************************************************************
                            0996     ;
                            0996     BUILD_CLUST:
03 00000119'EF    02   E3   0996          BBCS    #FLG_V_MKFORMED,FLAGS,10$ ; CONT IF CLUS_MASK NOT YET FORMED
            00A5  31   099E          BRW     BUILD_CLUSTX              ; MASK ALREADY FORMED; JUST EXIT
                            09A1     10$:
   00000111'EF    55   90   09A1          MOVB    R5,CLUS_MASK              ; BUILD
   00000112'EF    54   90   09A8          MOVB    R4,CLUS_MASK+1            ; .. CLUSTER
   00000113'EF    53   90   09AF          MOVB    R3,CLUS_MASK+2            ; ...... MASK
   00000114'EF    52   90   09B6          MOVB    R2,CLUS_MASK+3            ; ..............
                            09BD     ;
                            09BD     ; THE FOLLOWING CODE SETS CLUSTER A EQUAL TO CLUS_MASK
                            09BD     ;
              58  5A   D0   09BD          MOVL    R10,R8                    ; ESTABLISH FIRST EFN (EVENT FLAG NO.)
                  59   D4   09C0          CLRL    R9                        ; INIT OFFSET INTO CLUS_MASK
                            09C2     20$:
3A 00000111'EF    59   E0   09C2          BBS     R9,CLUS_MASK,30$          ; ISSUE $SETEF IF BIT FOR THIS FLAG IS SET
                            09CA          $CLREF_S EFN=R8                   ; ... OTHERWISE, ISSUE $CLREF
           68 50       E8   09D3          BLBS    R0,40$                    ; IF NORMAL STATUS, PROCESS NEXT EVENT FLAG
                            09D6          SS_CHECK NORMAL                   ; USE SS_CHECK TO TERMINATE TEST MODULE
                            0A04     30$:
                            0A04          $SETEF_S EFN=R8                   ; SET CURRENT EVENT FLAG
           2E 50       E8   0A0D          BLBS    R0,40$                    ; IF NORMAL STATUS, PROCESS NEXT EVENT FLAG
                            0A10          SS_CHECK NORMAL                   ; USE SS_CHECK TO TERMINATE TEST MODULE
                            0A3E     40$:
```

```
                        58   B6  0A3E   736           INCW     R8                     ; GET NEXT EFN
        FF7C 59    01   1F   9D  0A40   737           ACBB     #31,#1,R9,20$          ; GO DO NEXT EVENT FLAG
                                 0A46   738 BUILD_CLUSTX:
                             05  0A46   739           RSB                             ; RETURN TO CALLER
                                 0A47   740           .END
```

```
$$$$              = 000008BF R     04    COND5_T              000002B6 R     03
$$$CHARS          = 00000034             COND5_TAB            000002B6 R     03
$$$CHARS1         = 00000023             CONFLICT             ********    X  04
$$$CHARS2         = 00000024             CREPRN               00000051 R     02
$$$CHARS3         = 00000000             CTL$GL_PHD           ********    X  04
$$$CHARS4         = 00000000             DESC               = 00000010  G
$$$CHARS5         = 00000000             DIB$K_LENGTH       = 00000074
$$$COND_A         = 00000001             DIB$W_UNIT         = 0000000C
$$$STRINGS        = 00000001             EFLAG                ********    X  04
$$$STRINGS2       = 00000005             EFN                  000002AE R     03
$$T1              = 00000001             EXPV                 ********    X  04
$$T2              = 00000004             FAO_DESC             ********    X  04
ASCTOT              0000010C R     03    FAO_LEN              ********    X  04
BUILD_CLUST         00000996 R     04    FLAGS                00000119 R     03
BUILD_CLUSTX        00000A46 R     04    FLG_V_CLAOTHEV     = 00000000
BYTE              = 00000001  G          FLG_V_CLAPROC      = 00000001
CFLAG               ********    X  04    FLG_V_MKFORMED     = 00000002
CHMRTN              ********    X  04    FORM_CONDS           0000018E RG    04
CHM_CONT            ********    X  04    FORM_CONDSX          0000028F R     04
CLUS_MASK           00000111 R     03    IMAGNAM              0000008D R     02
CLUS_NAME_A         00000065 R     02    IO$_READVBLK         ********    X  04
CLUS_NAME_B         00000079 R     02    LONG               = 00000004  G
CLUS_STATE          00000115 R     03    MBXBUFF              0000008C R     03
COMP_SC             ********    X  04    MBXCHAN              00000008 R     03
COND1               0000016B RG    04    MBXCHANINFO          0000000C R     03
COND1_C           = 00000004             MBXUNIT              00000088 R     03
COND1_CLEANUP       0000016C RG    04    MOD_MSG_CODE         ********    X  04
COND1_E             0000013C R     03    MOD_MSG_PRINT        ********    X  04
COND1_H             0000011F RG    03    MSG1_INP_CTL         00000019 R     02
COND1_T             0000011A R     03    MSG3_ERR_CTL         00000039 RG    02
COND1_TAB           00000120 R     03    MSG_A                ********    X  04
COND2               0000016D RG    04    MSG_B                ********    X  04
COND2_C           = 00000000             MSG_CTXT             ********    X  04
COND2_CLEANUP       0000016E RG    04    MSG_DATA1            ********    X  04
COND2_E             00000204 R     03    NOTARG             = 00000000  G
COND2_H             0000015D RG    03    NULL               = 00000014  G
COND2_T             00000144 R     03    ONES                 ********    X  04
COND2_TAB           0000015E R     03    OTHER_EFN            0000010D R     03
COND3               0000016F RG    04    OUTPUT_MSG           ********    X  04
COND3X              00000188 R     04    PCV                  ********    X  04
COND3_C           = 00000000             PERM                 0000013C R     03
COND3_CLEANUP       00000189 RG    04    PHD$Q_PRIVMSK      = 00000000
COND3_E             00000255 R     03    PQL$_BYTLM         = 00000003
COND3_H             00000228 RG    03    PQL$_CPULM         = 00000004
COND3_T             00000204 R     03    PQL$_FILLM         = 00000006
COND3_TAB           00000229 R     03    PQL$_LISTEND       = 00000000
COND4               0000018A RG    04    PQL$_PGFLQUOTA     = 00000007
COND4_C           = 00000004             PQL$_PRCLM         = 00000008
COND4_CLEANUP       0000018B RG    04    PQL$_TQELM         = 00000009
COND4_E             000002AE R     03    PRIVMASK             00000000 R     03
COND4_H             0000025C RG    03    PRIV_ARGS          = 00000002
COND4_T             00000258 R     03    PROCESS_ERR          ********    X  04
COND4_TAB           0000025D R     03    QUAD               = 00000008  G
COND5               0000018C RG    04    QUOTALIST            000000AC R     02
COND5_C           = 00000014             RECV                 ********    X  04
COND5_CLEANUP       0000018D RG    04    REST_REGS            ********    X  04
COND5_H             000002B6 RG    03    SAVE_REGS            ********    X  04
```

```
                                                        M 15
SATSSS50                    SATS SYSTEM SERVICE TESTS  $ASCEFC (SUCC 16-SEP-1984 00:56:45  VAX/VMS Macro V04-00          Page  21
Symbol table                                                          5-SEP-1984 04:32:01  [UETPSY.SRC]SATSSS50.MAR;1          (1)

SS$_NORMAL                   ********    X   04
SS$_WASCLR                   ********    X   04
SUCCESS                      ********    X   04
SYS$ASCEFC                   ********   GX   04
SYS$CLREF                    ********   GX   04
SYS$CMKRNL                   ********   GX   04
SYS$CREMBX                   ********   GX   04
SYS$CREPRC                   ********   GX   04
SYS$DACEFC                   ********   GX   04
SYS$DELMBX                   ********   GX   04
SYS$DLCEFC                   ********   GX   04
SYS$FAO                      ********    X   04
SYS$GETCHN                   ********   GX   04
SYS$HIBER                    ********   GX   04
SYS$QIOW                     ********   GX   04
SYS$READEF                   ********   GX   04
SYS$SETEF                    ********   GX   04
SYS$SETPRN                   ********   GX   04
SYS$SETPRV                   ********   GX   04
SYS$WAKE                     ********   GX   04
TESTNUM                      ********    X   04
TEST_MOD_NAME                00000000 RG     02
TEST_MOD_NAME_D              00000009 R      02
TEST_MOD_SUCC                ********    X   04
TMD_ADDR                     ********    X   04
TM_CLEANUP                   00000159 RG     04
TM_SETUP                     00000000 RG     04
VERIFY                       00000290 RG     04
VERIFYX                      0000094C R      04
VFY_CLEANUP                  0000094D RG     04
WORD                       = 00000002  G
WRITE_MSG2                   ********    X   04

                                      +-----------------+
                                      ! Psect synopsis !
                                      +-----------------+

PSECT name                  Allocation            PSECT No.  Attributes
----------                  ----------            ---------  ----------
.  ABS  .                   00000000  (      0.)  00 (   0.)  NOPIC    USR    CON    ABS    LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                       00000000  (      0.)  01 (   1.)  NOPIC    USR    CON    ABS    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
RODATA                      000000CF  (    207.)  02 (   2.)  NOPIC    USR    CON    REL    LCL NOSHR NOEXE   RD   NOWRT NOVEC LONG
RWDATA                      000002B7  (    695.)  03 (   3.)  NOPIC    USR    CON    REL    LCL NOSHR NOEXE   RD     WRT NOVEC LONG
SATSSS50                    00000A47  (   2631.)  04 (   4.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD     WRT NOVEC BYTE

                                    +------------------------+
                                    ! Performance indicators !
                                    +------------------------+

Phase                   Page faults    CPU Time        Elapsed Time
-----                   -----------    --------        ------------
Initialization                   32    00:00:00.04     00:00:00.35
Command processing              111    00:00:00.68     00:00:01.72
Pass 1                          302    00:00:09.72     00:00:17.19
Symbol table sort                 0    00:00:00.65     00:00:00.73
Pass 2                          153    00:00:02.64     00:00:03.22
Symbol table output              18    00:00:00.12     00:00:00.12
```

```
Psect synopsis output            2      00:00:00.03     00:00:00.03
Cross-reference output           0      00:00:00.00     00:00:00.00
Assembler run totals           620      00:00:13.89     00:00:23.37

The working set limit was 1500 pages.
52689 bytes (103 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 370 non-local and 89 local symbols.
740 source lines were read in Pass 1, producing 30 object records in Pass 2.
51 pages of virtual memory were used to define 41 macros.

                                              +--------------------------+
                                              ! Macro library statistics !
                                              +--------------------------+

Macro library name                            Macros defined
------------------                            --------------
_$255$DUA28:[SHRLIB]UETP.MLB;1                       10
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                        1
_$255$DUA28:[SYSLIB]STARLET.MLB;2                    27
TOTALS (all libraries)                               38

778 GETS were required to define 38 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:SATSSS50/OBJ=OBJ$:SATSSS50 MSRC$:SATSSS50/UPDATE=(ENH$:SATSSS50)+EXECML$/LIB+SHRLIB$:UETP/LIB
```

0423 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY